Improved Localization Accuracy using Machine Learning: Predicting and Refining RSS Measurements

Cam Ly Nguyen[1,2], Orestis Georgiou[3,4], and Vorapong Suppakitpaisarn[2]

[1]Network System Laboratory, Corporate R&D Center, Toshiba Corporation, Japan
[2]Graduate School of Information Science and Technology, The University of Tokyo, Japan
[3]School of Mathematics, University of Bristol, UK
[4]Ultrahaptics, The West Wing Glass Wharf, Bristol, UK

***Abstract*—Wireless localization methods are often subject to errors due to radio signal fluctuations that are used to estimate inter-device separation distances. We propose a novel method called *MLRefine* to counter these effects by *refining* RSS measurement data to obtain more accurate values that can enhance ranging and localization accuracies. MLRefine uses machine learning methods to model the relationship between accurate values and features extracted from in silico RSS values. MLRefine then applies the trained model to features extracted from real RSS measurement values to return a predicted set of refined RSS values. The refined RSS values are shown through computer simulations and real experiments to improve localization accuracy.**

***Keywords* Wireless localization, measurement refinement, machine learning, wireless sensor networks, IoT networks.**

I. Introduction

Wireless sensor networks (WSNs) comprise a set of low-power wireless devices called sensor nodes with sensing and transceiving capabilities. These devices are typically tasked with sensing attributes of the surrounding environment, such as temperature, sound, pressure, humidity, etc. The data collected is then wirelessly funnelled through the network to a backhaul server for storing and processing. WSNs are an integral part of the Internet of Things (IoT) and are already being extensively used in environmental, military, civilian and industrial applications among many others. Importantly, WSNs are one of the main components of the Smart City future vision [1]. However, since geolocation information is often vital for the insightful processing and actuation of sensed data, a Global Positioning System (GPS) module is embedded to each wireless sensor. This luxury however adds significant production costs and power requirements (about 30mA at 3.3V) yet does not work when deployed in indoors. To alleviate such problems, several radio frequency (RF) based localization methods have been developed to estimate the sensor nodes locations. However, these too can have disadvantages such as localization errors due to noise and multipath effects. In this paper, we will leverage machine learning algorithms that can refine noisy RF data measurements and improve localization accuracy.

II. Background Intuition

RF localization methods can leverage proximitity and network connectivity [2], or may calculate the Angle of Arrival (AoA) or Time of Arrival (ToA), or Received Signal Strength (RSS) of incoming signals and use that to locate their location relative to some anchor points [3]. Compared with ToA, and AoA based methods, which require specific hardware, RSS based localization methods are more suitable for most IoT networks due to their simplicity and small energy and cost footprint implementation requirements. We will therefore only consider localization methods using RSS for ranging.

Most range based RSS localization methods require a ranging function f that relates the distance d between two wireless devices with the RSS value r. This can be expressed in dBm through Friis equation [4] as

$$r = f(d) + X = P_0 - 10\eta \log_{10} d + X \tag{1}$$

where P_0 is a known reference power value at distance of 1 meter from the transmitter. η is the path loss exponent usually set to 2 for free space wireless propagation or can be experimentally fitted via linear fitting (cf. Fig. 1a)). X is a random variable that captures the statistics of the RSS spatio-temporal fluctuations and therefore the inherent error of each data measurement. X is normally assumed as a zero-mean Gaussian random variable (in dB), i.e. $X \sim \mathcal{N}(0, \sigma^2)$.

Non-zero values of X cause ranging error and thus localization error. The Cramer-Rao lower bound (CRLB) can give a lower bound on the covariance of localization error for a given environment is proportional to the ratio σ/η [5]. More generally, the authors have previously shown theoretically and experimentally that the accuracy of localization matching algorithms scales with the ratio η/σ [6]. Hence, the lower the σ/η ratio is, the higher localization accuracy a localization estimator can achieve.

It follows that it would be useful to predict and remove the value of the random noise variable X attached to each measurement, thus decreasing the variance σ^2 and thereby improving the localization accuracy. Without information on surrounding environment, this is not trivial since it is unknown which factors affect a single signal. To achieve this, we start from the realization that RSS measurements between different node pairs in a network correlate with each other. We therefore extract useful geometrical information from non-local RSS measurements that when combined with local RSS measurements can generate a refined predicted set of more accurate local RSS measurements. To this end, we employ

standard regression techniques trained on computer simulated RSS data (in order to reduce training data collection time and cost) to build a prediction model, and then apply the model to real RSS data to achieve more accurate RSS measurements (here called *refined RSS*). The refined RSS data is general and can be used with any existing RSS location estimator to improve its accuracy. To the best of our knowledge, this has not been proposed before. Our main contributions are:

- Propose a novel method called *MLRefine* reducing the magnitude of noisy factors attached to RSS values.
- Substantiate MLRefine through various simulations and experiments.
- Quantify the impact of MLRefine by combining with state-of-the-art localization estimators.

III. Motivation and Related Work

Many range-based localization techniques consist of three phases: 1) the *calibration phase* during which the propagation model is calibrated using real RSS measurements (cf. Figure 1a), 2) the *measurement phase* during which nodes broadcast packets that are used at the receiver end to estimate and collect RSS values between all nearby nodes, and 3) the *localization phase* during which the collected RSS values are used to estimate each node's location by means of the propagation model and a localization algorithm. The proposed MLRefine method sits between phases 2) and 3). This section discusses related work on noise reduction methods (Section III-A) and gives background on localization algorithms used to quantify the impact of MLRefine (Section III-B).

A. Noise reduction methods

Types of ranging noise can be classified into outlier noise and normal noise [7]. Outlier, which is often caused by malicious attacks, non-line-of-sight (NLoS), hardware malfunction, means abnormal measurements far beyond the normal range. There are numerous attempts that try to detect distance outliers whose distance measurement errors are significantly large and hence are most likely outliers (see [8] for a survey). For instance, Jian et al. [7] use triangle inequality to detect outlier distances that are a factor of ten away from accurate measurements. Xiao et al. [9] propose method to detect outlier distances and outlier nodes that can collude due to malicious attacks. Recently, Xiao et al. [10] use multi-norms regularized matrix completion to realize localization methods that are robust to outlier noise. All of these methods consider that normal distance measurements are highly accurate or contain a small Gaussian error. Therefore, they are inapplicable to RSS-based localization in which normal RSS measurements contain large error. Considering RSS measurements, the authors [6] use a graph partitioning method to detect and remove NLoS signals that travel through walls.

Normal noise, which is often caused by multipath fading, shadowing, etc, is hard to be detected. Smoothing consecutive RSS measurements between two wireless nodes can reduce the normal noise. For instance, averaging RSS values from multiple antennas can reduce the effect of shadowing [11]. Time-consecutive RSS measurements can be filtered to reduce noise using particle filters such as Hampel filter, Kalman filter [12], or using averaging technique [13]. These methods, however, can only be realizable if there are transmissions between a pair of nodes.

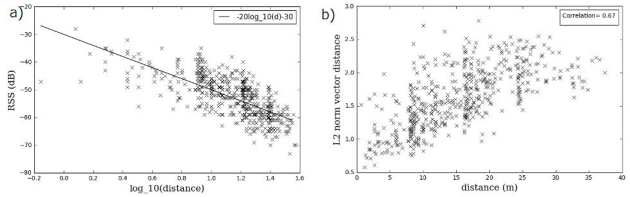


Fig. 1. a) Linear fitting of the path loss exponent η and reference power P_0 via real measurements. b) Correlation between RSS vectors similarity (here used Euclidean norm distance) and nodes separation distance.

The proposed method, MLRefine, is neither an outlier detection method nor a smoothing method. It uses correlation between RSS measurements between different pairs of nodes to reduce normal noise. It, however, can also be combined with the above techniques to further reduce the effect of noise. For instance, outlier RSS measurements are first detected and removed using [6]. A smoothing method such as averaging is then applied to reduce the effect of shadowing. MLRefine is, finally, applied to further reduce normal noise level.

B. Cooperative localization algorithms

Once the data is refined from noisy errors, a localization algorithm must be applied. In contrast to traditional multilateration techniques [14], cooperative localization methods estimate all node positions simultaneously using measurements between almost all nodes rather than localizing each node, thus enhancing accuracy [5]. There are many cooperative localization algorithms such as multidimensional scaling (MDS), semidefinite programming (SDP) [15], stochastic optimization (e.g. simulated annealing (SA)) [3], and matching methods [6]. Among cooperative localization, SDP localization enhance accuracy and can provide deterministic solutions, while localization matching technique enhance accuracy when a set of node positions is provided. To quantify the impact of proposed MLRefine, we will, therefore, employ SDP method designed for resolving RSS-base localization problems (i.e., Formulas (11) and (12) in Ref. [15]) and localization matching method called MLMatch that is described in Sections VI.A, VI.B.3 of Ref. [6]. These localization algorithms will be applied to the refined datasets in Sections V and VI.

IV. The MLRefine Algorithm

A. Problem Definition

Consider a WSN composed of m nodes that are labelled $1, 2, ..., m$ and equipped with a radio transceiver such that they can exchange messages with each other. RSS measurements between nodes can be obtained and thus sent to a backhaul server for post-processing. The server then saves all the RSS values in a square $m \times m$ matrix $\mathcal{R}$ with entries $r_{i,j}$ equal to RSS value between nodes i, j. If some RSS values are missing, their corresponding entries are saved as null values. This network model for localization is often used in literatures

[5], [15]. MLRefine is a method that enable the server to refine these measurements and output a refined RSS square matrix $\mathcal{R}'$ with refined RSS entries $r'_{i,j}$ that can provide more accurate ranging and therefore can improve localization estimators. In other words, $r'_{i,j}$ is closer to $\tilde{r}_{i,j} = f(d_{i,j})$ (here called *non-noisy RSS value*, cf. Formula (1)) than $r_{i,j}$.

B. MLRefine Intuition

MLRefine exploits the inherent geometrical properties of the network deployment space. Namely, since wireless sensor nodes are located in Euclidean space, the pair-separation-distance between two nodes correlates strongly with various distances of other node pairs. Note that the latter can be estimated via RSS values. In practice, if nodes i and j are close to each other, then the similarity between RSS values $r_{i,k}$ and $r_{j,k}$ is high for an arbitrary node k. In contrast, if nodes i and j are far from each other, then the similarity between $r_{i,k}$ and $r_{j,k}$ is low for some node k. It follows that inter-node separation distance $d_{i,j}$, and therefore non-noisy RSS value $\tilde{r}_{i,j}$, are correlated with some similarity metric between RSS vectors $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$, where $\boldsymbol{r_i} = [r_{i,1}, r_{i,2}, \ldots, r_{i,m}]$. Figure 1 b) illustrates the strong correlation between distances $d_{i,j}$ and the similarity of experimentally measured RSS vectors. Here, we used Euclidean norm distance the of two vectors $\boldsymbol{r_i}$ and $\boldsymbol{r_j}$ as the similarity metric. As a result we may conclude that RSS vector similarities can be used as features to predict non-noisy RSS values. However, the correlations between RSS vector similarities and non-noisy RSS values are not known *a priori* and are difficult to formulate. To avert this problem, MLRefine leverages standard machine learning regression methods that encode said mathematical relationships using computer generated training data.

C. The MLRefine Algorithm

The proposed localization method is illustrated in Figure 2. The whole process is schematically broken down into four blocks: the propagation model calibration block, the training block, the refining block, and finally the localization algorithm. The calibration process will not be discussed since it can be chosen from any existing method such as linear fitting method (cf. Fig. 1). We also will not discuss the final block in detail since MLRefine is to a large extent agnostic to the chosen algorithm and should improve localization accuracy and performance regardless by decreasing the ratio σ/η.

1) Training phase: The training phase, described by Algorithm 1, is designed to study the relationship between a non-noisy RSS value and features extracted from other noisy RSS values. As such, a key benefit of MLRefine is that it can be trained *in silico* (i.e., using computer simulated data). First, m pairs of coordinates are generated at random from some finite subspace $\mathcal{A}$ of $\mathbb{R}^2$ of comparable area to that of the intended WSN deployment. These coordinates represent the m sensor node locations. Inter-node separation distances $d_{i,j}$ are then calculated and converted into non-noisy RSS values $\tilde{r}_{i,j}$ using the Propagation Model (1). Secondly, Gaussian noise is then added to all the non-noisy RSS values $r_{i,j} = \tilde{r}_{i,j} + X$ and stored in a symmetric $m \times m$ matrix $\mathcal{R}$. In our simulations we have chosen the standard deviation of the added noise from the range of $\sigma \in [3, .., 7]$ dB as suggested from experiments [6]. Note that we are neither restricted to Gaussian noise nor specific range of σ; for instance in environments where distribution of X can be obtained accurately, it can be substituted for the chosen values of X. In the third step, *features* corresponding to each node pair i, j are then extracted. In machine learning, a feature is any measurable property or characteristic of a phenomenon being observed. Choosing informative and discriminating features is a crucial step for effective algorithms in regression. Whilst there are learning techniques that can extract features automatically from raw datasets such as image, voice, etc. (e.g., CNN), efficient learning techniques for treating weighted graph data (i.e. wireless network data) is still unknown. Moreover, efficient learning from artificial data (in this case simulated RSS data) is still a challenge. We therefore, heuristically extract features described in Table I, in which vector similarities are intuitively correlated with non-noisy RSS values (cf. Section IV-B). Namely, we extract the Pearson correlation, the average absolute and Euclidean norms. Other statistical values such as the standard deviation of vectors $\boldsymbol{r_i}$ and $\boldsymbol{r_j}$ and combinations of them are also extracted. All of these values are not strongly dependent on a single RSS value but rather represent the whole network thus alleviating negative influence caused by the difference between a single simulated RSS value and real or experimental RSS one. Note that this is not a unique set of features since other statistical values could be used just as well.

TABLE I
DESCRIPTION OF FEATURES

Feature type	Feature name	Formula
Vector similarity	Pearson correlation	$\frac{\mathrm{cov}(\boldsymbol{r_i}, \boldsymbol{r_j})}{\sigma_{\boldsymbol{r_i}} \sigma_{\boldsymbol{r_j}}}$
	Average Absolute value norm	$\frac{\lVert \boldsymbol{r_i} - \boldsymbol{r_j} \rVert_1}{\lVert \boldsymbol{r_i} \rVert_0}$
	Average Euclidean norm	$\frac{\lVert \boldsymbol{r_i} - \boldsymbol{r_j} \rVert_2}{\lVert \boldsymbol{r_i} \rVert_0}$
Other statistic values	Standard deviation of $\boldsymbol{r_i}, \boldsymbol{r_j}, \boldsymbol{r_i} - \boldsymbol{r_j}, \boldsymbol{r_i} + \boldsymbol{r_j}$ Average of $\boldsymbol{r_i}, \boldsymbol{r_j}$	
Raw RSS value	RSS value between 2 nodes	$r_{i,j}$

where $\lVert \boldsymbol{x} \rVert_p = (|x_1|^p + |x_2|^p + \ldots)^{1/p}$,
$cov(\boldsymbol{x}, \boldsymbol{y})$ is covariance of $\boldsymbol{x}$ and $\boldsymbol{y}$, $\sigma_{\boldsymbol{x}}$ is standard deviation of $\boldsymbol{x}$

Once features have been selected and extracted from the noisy RSS matrix $\mathcal{R}$, we then use regression techniques to model the relationship between the non-noisy RSS values $\tilde{r}_{i,j}$ and its corresponding features. There are a number of standard regression techniques suitable for this task. We ran linear regression, support vector machine, neural networks with various parameter settings, then compared the results on some sample sets. The results showed that except linear regression, other regression models gave similar results. This is because, the number of features is small. In the remaining of this paper, we will report results of using a simple neural network model with a single hidden layer consisting of 10

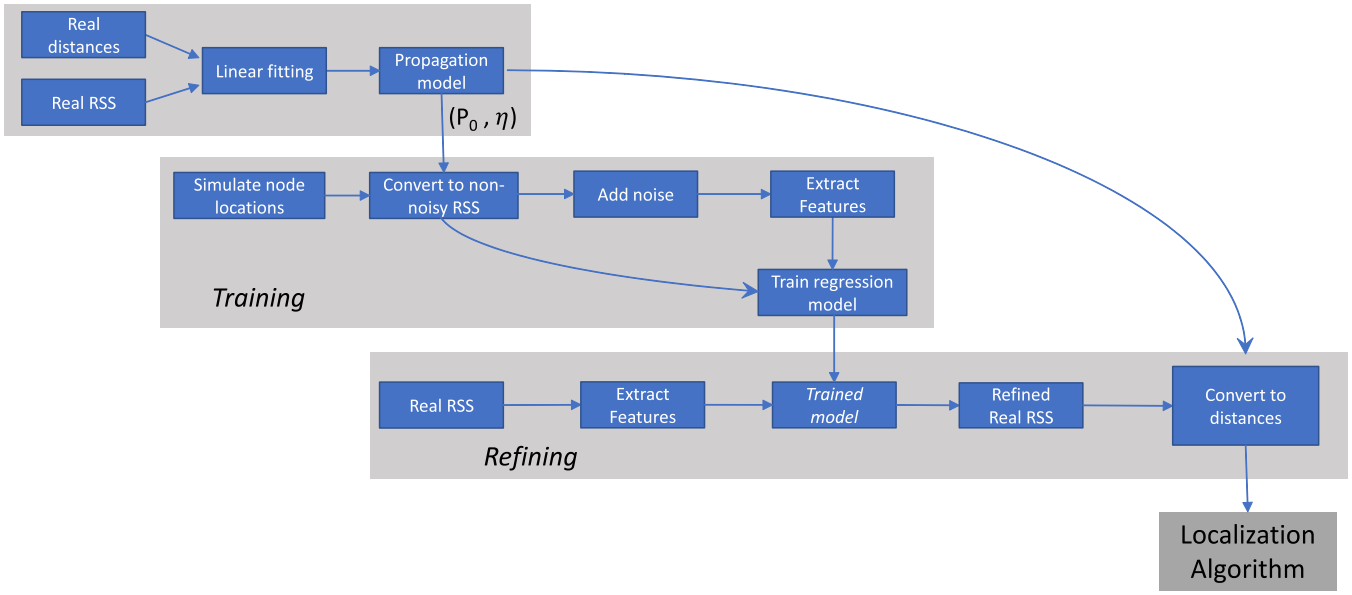


Fig. 2. Schematic showing how different processes of MLRefine are inter-linked in each of the three blocks leading towards an localization estimator algorithm.

hidden nodes, and activation function of ReLu.

Algorithm 1 Training phase

Input: $r = f(d)$ {propagation model}, m {number of nodes}, $\mathcal{A}$ {nodes deployment area}

Output: F {trained model}

Training data generation

for each $\sigma \in \boldsymbol{\sigma}$ { $\boldsymbol{\sigma}$ is a bounding range of σ } **do**
  Generate m random coordinates in Area $\mathcal{A}$
  for each $i, j \in \{1, \ldots, m\}$ **do**
    $\tilde{r}_{i,j} \leftarrow f(d_{i,j})$ {non-noisy RSS value}
    $r_{i,j} \leftarrow \tilde{r}_{i,j} + X$, where $X \sim \mathcal{N}(0, \sigma^2)$ {noisy RSS value}
  end for
Feature Extraction
  for each $i, j \in \{1, .., m\}$ **do**
    $\boldsymbol{features_{i,j}} \leftarrow$ features described in Table I
  end for
end for
Training
Model Function F satisfying $\tilde{r}_{i,j} \approx F(\boldsymbol{features_{i,j}}), \forall i, j$, using a regression method.

2) Refining phase: The refining phase of MLRefine is described by Algorithm 2 and begins after a backhaul server has collected real RSS measurements.To predict an accurate RSS value corresponding to nodes i, j, MLRefine first extracts features from real RSS values. This process is similar to the feature extraction process that is described in Section IV-C1. MLRefine then applies the trained model F obtained from the training phase to the extracted features to yield the refined value $r'_{i,j}$. The refined RSS values can be applied with the localization matching algorithm [6], or can be converted to distances using the propagation model (1) before the SDP

Algorithm 2 Refining phase

Input: $\mathcal{R}$ {measured RSS matrix with entries $r_{i,j}$}, F {trained model}

Output: $\mathcal{R}'$ {refined RSS value matrix with entries $r'_{i,j}$}

for each $i, j \in \{1, \ldots, m\}$ **do**
  $\boldsymbol{features_{i,j}} \leftarrow$ features described in Table I
  $r'_{i,j} \leftarrow F(\boldsymbol{features_{i,j}})$
end for

localization algorithm [15] is applied to estimate nodes' locations. Alternatively, the refined RSS dataset can be used in any off-the-shelf range based localization algorithm to improve multilateration accuracy.

V. MLRefine Performance Evaluation via Simulations

In order to substantiate our proposed method, we have performed three computer simulations. We compare the variance fluctuations of raw RSS values against the fluctuations of refined RSS values, and we also contrast the final localization accuracy achieved when SDP localization method is applied (see Section III-B).

A. Parameter settings

Simulation 1: To test our proposed method against 1) different numbers of nodes $m \in \{10, ..., 100\}$ are deployed randomly in a 100×100m square domain, and 2) different noise distributions X. In each random realization and for each pair of nodes, we generate two separate pairs of RSS values using Formula (1); one with a Gaussian distributed $X \sim \mathcal{N}(0, \sigma^2)$, where $\sigma = 5.57$ and one with a Rayleigh distributed X whose probability density function is given by

$$f_X(x) = \lambda 10^{x/10} \exp\left(-\lambda 10^{x/10}\right) \ln 10/10, \qquad (2)$$

TABLE II
DESCRIPTION OF PARAMETERS IN SIMULATIONS

	Simulation 1	Simulation 2	Simulation 3
# of Nodes	10 - 100	49	50
Area(m)	100×100	100×100	C-shape
Layout	Random	Random&Grid	Random
X distribution	Gaussian&Rayleigh	Gaussian	Gaussian&Rayleigh
Std of X	5.57	3.0 - 7.0	5.57
# of anchors	-	4	10 - 25

where $\lambda = 0.561$, in which case the standard deviation of X is $\sigma \approx 5.57$ [6]. *Simulation 2*: To test MLRefine against different types of node distributions and different values of Gaussian noise variance we generate 49 nodes in a 100×100m square domain under two types of deployment layouts: 1) a random layout, and 2) a 7×7 grid layout with one grid length equals to 14m. In each random realization and for each pair of nodes, random RSS values were generated using (1) with $X \sim \mathcal{N}(0, \sigma^2)$ and for different values of σ chosen uniformly from $\{3, 3.5, 4, 4.5, \ldots 7\}$. As discussed in Section II, reducing the variation of signal fluctuation equivalent to reducing the lower bound of the variance of localization error. However, to quantify the impact of MLRefine, we feed SDP localization method, which is known as enhancing accuracy [15], with refined and unrefined (i.e. raw) RSS data. *Simulation 3*: To test our proposed method's performance in anisotropic domain shape deployments against 1) different noise distributions: Gaussian and Rayleigh distributions given by Formula (2), and 2) different numbers of anchor nodes: from 20% to 50% of all the nodes, we generate 50 nodes randomly deployed in a C-shape domain (Figure 3 3-c). In this instance, we use the SDP method with $a \in \{10, 12, 15, \ldots, 25\}$ anchors located randomly in the C-shaped WSN deployment region. All RSS values in the above three simulations were generated using common parameters $P_0 = -30$ dB and $\eta = 2$. The values of all other parameters used are summarized in Tab. II For each set of parameters, simulations were performed 10 times to obtain statistical averages.

B. Results and Analysis

In every simulation, we first run the proposed MLRefine algorithm and measure the accuracy of the output refined values $r'_{i,j}$ compared to non-noisy RSS values. Namely, we measure the standard deviation σ' of variable X' , where $X' = r'_{i,j} - P_0 + 10\eta \log_{10} d_{i,j}$. Note that we calculate standard deviation of X' under the assumption that the mean of X' is zero, which is equivalent to the mean squared error of refined RSS values. We then compare σ' with σ, which is the standard deviation of input random variable X of raw (unrefined) RSS values. As discussed in Section I, we expect that the smaller of standard deviation of the RSS values used for localization, the better the accuracy that a localization estimator can achieve. This is why it is meaningful here to compare σ' with σ. In addition to the fluctuations, we then also apply the SDP localization method and compare the corresponding localization errors using raw RSS values and refined RSS values.

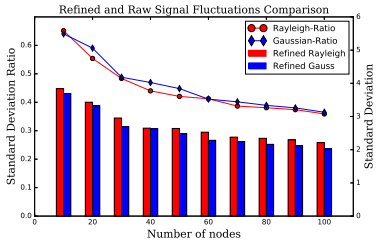
The results are plotted in Figure 3 indicating that in any case, the value of σ' is much smaller than σ. The results of Simulation 1 (Figure 3. 1-a) indicates that the standard deviation of refined RSS values σ' decreases while the standard deviation of raw (unrefined) RSS values σ is unchanged (equals to 5.57, cf. Section V-A), hence the ratio $\sigma'/\sigma < 1$. Also, a clear positive trend is seen as the number of sensor nodes m increases. Figure 3. 1-a) also validates the robustness of MLRefine against other random variable distribution models for wireless fading such as the Rayleigh distribution even though the training data is generated using a Gaussian distribution. Figures 3. 1-b, c) illustrate that the refined distribution of variable X' is similar to the distribution of X, namely X' also follows Gaussian (or Rayleigh) when X follows Gaussian (or Rayleigh), but with smaller variation. The results of Simulation 2 (Figures 3. 2) illustrate the localization error of SDP method using raw vs. refined RSS values when nodes are distributed randomly or in a grid layout. Observe that the localization error is much smaller for refined RSS data compared to the raw (unrefined) RSS values. Moreover, localization error significantly decreases when the standard deviation of raw RSS values σ decreases, which suggest robustness of MLRefine especially in noisy networks. The figures also validate the robustness of MLRefine against other node distribution patterns such as a grid structure, even though the nodes are distributed under a random uniform spatial distribution during the training phase. The results of Simulation 3 (Figures 3. 3) illustrate the localization error of SDP method using raw and refined RSS values when nodes are distributed in a C-shape domain. Observe that MLRefine improves localization error consistently in these settings too and is impervious to using different fading models for noise and fading effects X despite our training phase using Gaussian noise, with uniform node deployments in a square domain. Further improvements could be anticipated if the training data utilised anisotropic C-shaped deployment regions. In conclusion, the results illustrate that MLRefine is robust in various environments despite the difference of distribution models between training data and raw RSS values.

VI. EXPERIMENTAL INVESTIGATION

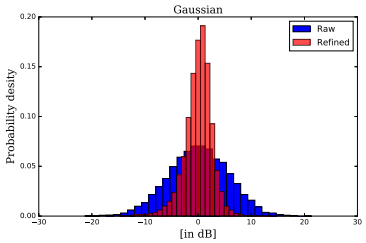
This section investigates and validates the performance benefits of MLRefine through real experiments. We compare the variance fluctuation of the experimentally measured raw RSS values against the fluctuation of the refined RSS values, and we also contrast the localization accuracy achieved when combined with MLMatch algorithm (see Section III-B). The accuracy of the algorithm is defined as the ratio of number of correctly matched nodes to the number of total nodes, which is different from other localization methods.

A. Description of Experimental Setup

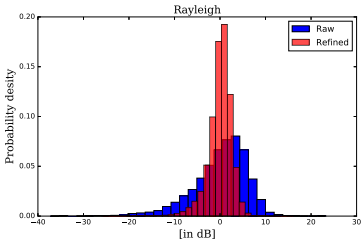
We have performed experiments in Toshiba premises in Japan, utilising 33 Toshiba wireless devices operating at 920MHz band. The devices were placed at specific positions



1-a) Sigma comparison

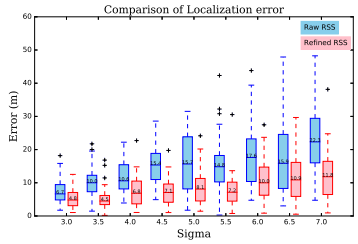


1-b) Histogram Gaussian

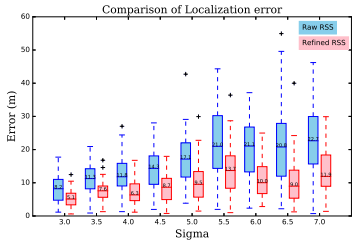


1-c) Histogram Rayleigh

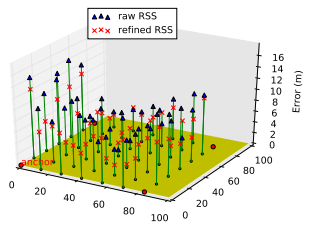
Simulation 1



2-a) Random layout

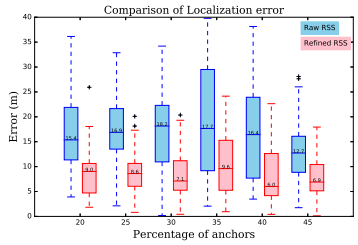


2-b) Grid layout

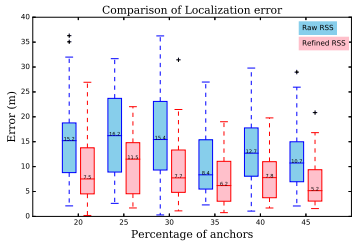


2-c) Error distribution ($\sigma = 3$)

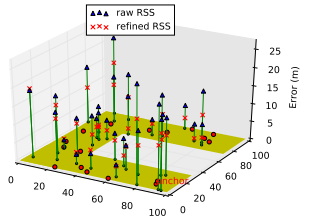
Simulation 2: Localization error comparison



3-a) Gaussian



3-b) Rayleigh



3-c) Error distribution (35% anchors)

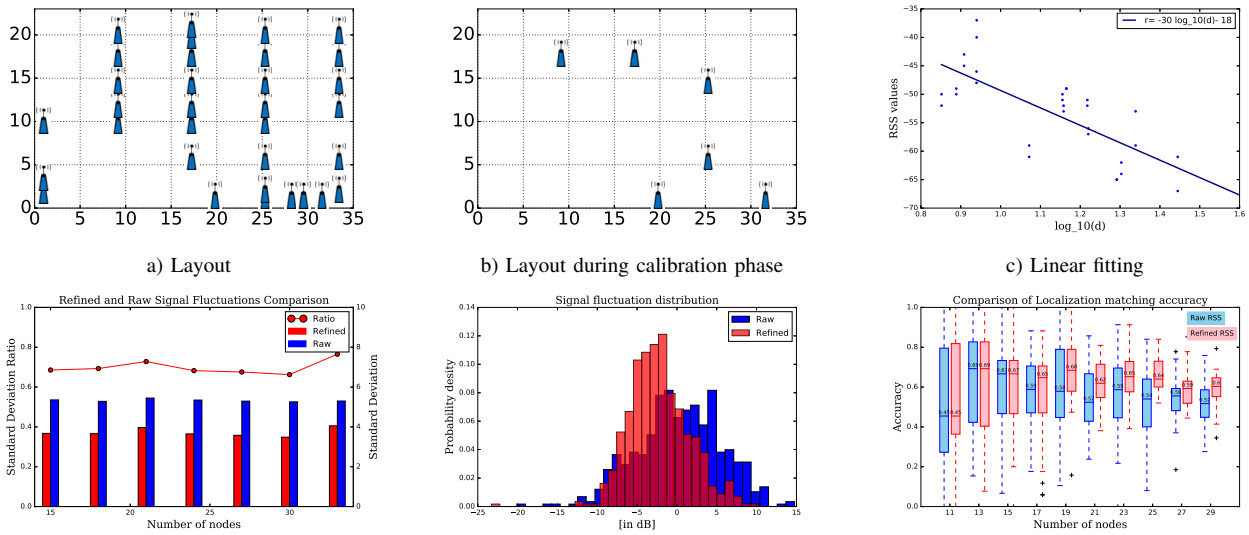
Simulation 3: Localization error comparison

Fig. 3. Simulation 1 results: 1-a) Comparison between fluctuations of raw (unrefined) RSS values (i.e. σ) and fluctuation of refined values (i.e. σ'). Lines express ratio σ'/σ, bars express values of σ'. 1-b, c) Probability density comparison between fluctuations of raw RSS values (blue) and refined values (red) under Gaussian distribution (1-b) and Rayleigh distribution (1-c). Simulations 2, 3 results: a, b) Comparison of localization error using SDP method between raw RSS values (in blue) and refined RSS value (in red). c) Illustration of localization error in grid distribution with $\sigma = 3$ (2-c) and C-shape domain with 17 anchors (3-c).

described in Figure 4a). For further details of the experimental setup, prototype, and surrounding environment, readers are referred to Experiment 5 in [6] from which much of the data was obtained. *Calibration:* In order to determine the propagation model, we put 6 wireless devices located at positions described in Figure 4b). The devices communicate with each other, thus RSS value between each pair of nodes can be collected and sent to a server. The server uses a linear fit to calculate parameters P_0 and η (Figure 4c).

B. Results and Analysis

Using data from a total of 33 wireless nodes we randomly select a subset and run MLRefine and compare refined RSS values with the raw (unrefined) ones. Training was performed through computer simulations that have the same parameters as the experimental setup. Similar to our simulation results and analysis, we measure the expected accuracy of refined values $r'_{i,j}$ through the standard deviation of their fluctuations σ'. In addition, we run a localization matching algorithm (MLMatch) on both RSS types: raw RSS values and refined RSS values. The results of this experiment are plotted in Figure 4d) indicating that the value of σ' is much smaller than σ. Figure 4d) depicts the distribution of variables X and X' illustrating that the distribution of unrefined (raw) signal fluctuations follows neither Gaussian nor Rayleigh distributions. This is due to the effect of the surrounding realistic indoor environment such as reflections and scattering. By truncating the original database we can study the experimental effects of MLRefine on smaller networks. We find that MLRefine can enhance the localization accuracy as seen in Figure 4e) which illustrates the performance gains of the localization algorithm MLMatch running on refined RSS datasets. These gains appear to onset when the number of nodes m is bigger than 17. For instance,



a) Layout b) Layout during calibration phase c) Linear fitting

d) Signal fluctuations comparison e) Histogram f) MLMatch accuracy

Fig. 4. Experimental setup and results: d) Comparison between fluctuations of raw RSS values (i.e. σ) and fluctuation of refined values (i.e. σ'). Lines express ratio σ'/σ, bars express values of σ(blue) and σ'(red). e) Probability density comparison between fluctuations of raw RSS values (blue) and refine values (red). f) Localization matching accuracy comparison using raw RSS values and refined RSS values for different number of nodes.

the difference of median is up to 10% for $m = 25$, up from 54% to 64%, respectively. This trend is similar to the results in Simulation 1 (cf. Figure 3a), where the efficiency of MLRefine increased with the number of wireless devices being used.

VII. Conclusions

We have proposed a method called *MLRefine* to refine raw RSS data collected from wireless sensor networks in order to improve wireless ranging and therefore localization accuracy. MLRefine uses machine learning algorithms to extract and exploit the inherent spatial network geometrical correlations that are hidden in noisy RSS datasets used for RF wireless localization. These correlations are captured by features that are then used to reduce the magnitude of RSS fluctuations, in turn improving the localization accuracy. We note that training data can be generated offline, thus minimizing the cost of collecting training data. We have validated the efficiency of MLRefine through various computer simulations but also real world experiments in Japan. Despite our initial encouraging results, there is many aspects that still need to be addressed. These include the consideration of large sparse networks, the effects due to realistic environments, and the adaptation of our algorithm to AoA and ToA datasets.

References

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[2] C. Nguyen, O. Georgiou, and Y. Doi, "Maximum likelihood based multihop localization in wireless sensor networks," in *2015 IEEE International Conference on Communications (ICC)*, pp. 6663–6668, IEEE, 2015.

[3] G. Mao, B. Fidan, and B. D. Anderson, "Wireless sensor network localization techniques," *Computer networks*, vol. 51, no. 10, pp. 2529–2553, 2007.

[4] A. Bose and C. H. Foh, "A practical path loss model for indoor wifi positioning enhancement," in *Information, Communications & Signal Processing, 2007 6th International Conference on*, pp. 1–5, IEEE, 2007.

[5] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal processing magazine*, vol. 22, no. 4, pp. 54–69, 2005.

[6] C. L. Nguyen, O. Georgiou, Y. Yonezawa, and Y. Doi, "The wireless localization matching problem," *IEEE Internet of Thing (IoT-J)*, pp. 1312–1326, 2017.

[7] L. Jian, Z. Yang, and Y. Liu, "Beyond triangle inequality: Sifting noisy and outlier distance measurements for localization," in *2010 Proceedings IEEE INFOCOM*, pp. 1–9, March 2010.

[8] H. Abukhalaf, J. Wang, and S. Zhang, "Outlier detection techniques for localization in wireless sensor networks: A survey," *International Journal of Future Generation Communication and Networking*, vol. 8, no. 6, pp. 99–114, 2015.

[9] Q. Xiao, K. Bu, Z. Wang, and B. Xiao, "Robust localization against outliers in wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 9, no. 2, p. 24, 2013.

[10] F. Xiao, W. Liu, Z. Li, L. Chen, and R. Wang, "Noise-tolerant wireless sensor networks localization via multinorms regularized matrix completion," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2409–2419, 2018.

[11] S. Hamdoun, A. Rachedi, and A. Benslimane, "Comparative analysis of rssi-based indoor localization when using multiple antennas in wireless sensor networks," in *Selected Topics in Mobile and Wireless Networking (MoWNeT), 2013 International Conference on*, pp. 146–151, IEEE, 2013.

[12] Z. Chen, H. Zou, H. Jiang, Q. Zhu, Y. C. Soh, and L. Xie, "Fusion of wifi, smartphone sensors and landmarks using the kalman filter for indoor localization," *Sensors*, vol. 15, no. 1, pp. 715–732, 2015.

[13] C. L. Nguyen and A. Khan, "Wilad: Wireless localisation through anomaly detection," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pp. 1–7, IEEE, 2017.

[14] Y. Zhou, J. Li, and L. Lamont, "Multilateration localization in the presence of anchor location uncertainties," in *Global Communications Conference (GLOBECOM), 2012 IEEE*, pp. 309–314, IEEE, 2012.

[15] P. Biswas, T.-C. Lian, T.-C. Wang, and Y. Ye, "Semidefinite programming based algorithms for sensor network localization," *ACM Transactions on Sensor Networks (TOSN)*, vol. 2, no. 2, pp. 188–220, 2006.